**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of: Richard D. Dettinger et al.	§	Filed: April 8, 2004
Serial No.: 10/821,228	§	Group Art Unit: 2161
Confirmation No.: 9498	§	Examiner: Jared M. Bibbee
For: METHOD AND SYSTEM FOR RELATIONSHIP BUILDING FROM XML	§	

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office to fax number 571-273-8300 to the attention of Examiner Jared M. Bibbee, or electronically transmitted via EFS-Web, on the date shown below:

October 2, 2007	/John C. Garza/
Date	John C. Garza

Dear Sir:

APPEAL BRIEF

Applicants submit this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 2161 dated April 2, 2007, finally rejecting claims 1-44. The final rejection of claims 1-44 is appealed. This Appeal Brief is believed to be timely since it is transmitted by the extended due date of October 2, 2007, as set by the filing of a Notice of Appeal on July 2, 2007.

Please charge the fee of $510.00 for filing this brief to:

Deposit Account No. 09-0465 / ROC920030347US1.

TABLE OF CONTENTS

Real Party in Interest

The present application has been assigned to International Business Machines Corporation, Armonk, New York.

Related Appeals and Interferences

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-44 are pending in the application. Claims 1-44 were originally presented in the application. Claims 1-44 stand finally rejected as discussed below. The final rejections of claims 1-44 are appealed. The pending claims are shown in the attached Claims Appendix.

Status of Amendments

All claim amendments have been entered by the Examiner. No amendments to the claims were proposed after the final rejection.

Summary of Claimed Subject Matter

Claimed embodiments include computer-implemented methods (see claims 1-18), computer programs stored on computer readable media (see claims 19-43), and an article (see claim 44) directed to relationship building from XML. More specifically, embodiments provide a method, system and article of manufacture for relationship building based on text-based markup languages and, more particularly, for representing relationships between data elements defined according to a first physical representation of data in a logical representation. The logical representation abstractly describes a second physical representation of the data, which is generated from the first physical representation of the data. *See Application*, page 8, lines 18-22; *Abstract.* In one embodiment, the first physical representation of the data is a document in text-based markup language, such as an XML document, and the second physical representation of the data is a relational database schema. See *Application*, page 8, lines 22-25. For a description of the physical environment of the invention, see *Application*, p. 11-14, for a description of the software environment of the invention, see *Application*, p. 14-19, and for a description of methods for relationship building from XML, see *Application*, p. 19-36.

A. CLAIM 1 - INDEPENDENT

A method of logically representing relationships between data elements defined according to a first physical representation of data. See *Application*, page 3, line 18 – page 4, line 2. As claimed, the method includes providing a logical representation of the data, the logical representation abstractly describing a second physical representation of the data, wherein the second physical representation of the data is generated from the first physical representation of the data. See *Application*, page 24, line 18 – page 25, line 12; FIG. 8. The method also includes, on the basis of the relationships between the data elements defined according to the first physical representation of the data, determining corresponding relationships between corresponding data structures defined according to the second physical representation of the data. See *Application*, page 25, line 13 – page 30, line 7; FIG. 8-9. The method also includes generating logical

relationships abstractly describing the determined corresponding relationships, each logical relationship defining a path between data structures of the second physical representation. See *Application*, page 25, line 19 – page 31, line 34; FIG. 8-9. The method also includes associating the generated logical relationships with the logical representation of the data. See *Application*, page 31, line 35 - page 33, line 22; FIG. 9. The method also includes storing the associations and the generated logical relationships on one or more computer-readable storage media. See *Application*, page 10, lines 13-27; page 31, line 35 - page 33, line 22; FIG. 9.

B. CLAIM 11 - INDEPENDENT

A method of logically representing relationships between data elements defined according to a first physical representation of data. See *Application*, page 4, lines 3-18. As claimed, the method includes generating a second physical representation of the data from the first physical representation. See *Application*, page 24, lines 18 – page 25, line 12; FIG. 8. The method also includes generating a logical representation of the data as represented according to the second physical representation, the logical representation abstractly describing the second physical representation of the data. See *Application*, page 24, lines 18 – page 25, line 12; FIG. 8. The method also includes, on the basis of the relationships between the data elements defined according to the first physical representation of the data, determining corresponding relationships between corresponding data structures defined according to the second physical representation of the data. See *Application*, page 25, line 13 – page 30, line 7; FIG. 8-9. The method also includes generating logical relationships abstractly describing the determined corresponding relationships. See *Application*, page 25, line 19 – page 31, line 34; FIG. 8-9. The method also includes including the generated logical relationships with the logical representation, wherein each generated logical relationships describes a path for traversing the second physical representation from a first data structure to a second data structure when processing a query requesting information related to the first and second data structures. See *Application*, page 31, line 35 - page 33, line 22; FIG. 9. The method also includes storing the second physical representation, the generated logical relationships, and the logical representation on

one or more computer-readable storage media. See *Application*, page 10, lines 13-27; page 31, line 35 - page 33, line 22; FIG. 9.

C. CLAIM 17 - INDEPENDENT

A method of logically representing relationships between data elements described in an eXtended Markup Language (XML) document. See *Application*, page 4, line 19 – page 5, line 2. As claimed, the method includes retrieving a relational database schema for a plurality of data structures, each data structure corresponding to one of the data elements. See *Application*, page 23, lines 1-12; FIG. 6. The method also includes retrieving a logical representation abstractly describing the relational database schema. See *Application*, page 24, line 18 – page 25, line 12; FIG. 8. The method also includes determining the relationships between the data elements from the XML document. See *Application*, page 25, lines 13-26; FIG. 8. The method also includes, on the basis of the determined relationships, determining corresponding relationships between corresponding data structures defined according to the relational database schema. See *Application*, page 25, line 13 – page 30, line 7; FIG. 8-9. The method also includes generating logical relationships abstractly describing the determined corresponding relationships. See *Application*, page 25, line 19 – page 31, line 34; FIG. 8-9. The method also includes including the generated logical relationships with the logical representation, where each of the generated logical relationships describes a path for traversing a relational database constructed according to the relational database schema from a first data structure to a second data structure when processing a query requesting information related to the first and second data structures. See *Application*, page 31, line 35 - page 33, line 22; FIG. 9. The method also includes storing the generated logical relationships on one or more computer-readable storage media. See *Application*, page 10, lines 13-27; page 31, line 35 - page 33, line 22; FIG. 9.

D. CLAIM 18 - INDEPENDENT

A method of querying physical data logically represented by a data abstraction model, wherein the physical data being queried is contained in data structures

generated from a data source having a different schema from the data structures containing the physical data being queried. See *Application*, page 5, lines 3-14. As claimed, the method includes receiving an abstract query comprising logical fields and corresponding values, wherein each of the logical fields is defined in the data abstraction model and wherein one or more of the logical fields are result fields to be returned by execution of the abstract query. See *Application*, page 13, lines 15-28; FIG. 2. The method also includes transforming, by operation of a processor, the abstract query into an executable query capable of being executed against the physical data; wherein the transforming is done using the data abstraction model and wherein the data abstraction model defines a specific path for traversing the data structures containing the physical data to reach the one or more result fields. See *Application*, page 13, line 29 – page 17, line 23; FIG. 2; page 25, line 19 – page 31, line 34; FIG. 8-9.

E. CLAIM 19 - INDEPENDENT

A computer-readable medium containing a program which, when executed by a processor, performs a process of logically representing relationships between data elements defined according to a first physical representation of data. See *Application*, page 5, lines 15-28. As claimed, the program contained in the computer readable medium includes retrieving a logical representation of the data, the logical representation abstractly describing a second physical representation of the data, wherein the second physical representation of the data is generated from the first physical representation of the data. See *Application*, page 24, line 18 – page 25, line 12; FIG. 8. The program contained in the computer readable medium also includes, on the basis of the relationships between the data elements defined according to the first physical representation of the data, determining corresponding relationships between corresponding data structures defined according to the second physical representation of the data. See *Application*, page 25, line 13 – page 30, line 7; FIG. 8-9. The program contained in the computer readable medium also includes generating logical relationships abstractly describing the determined corresponding relationships, each logical relationship defining a path between data structures of the second physical representation. See *Application*, page 25, line 19 – page 31, line 34; FIG. 8-9. The

program contained in the computer readable medium also includes associating the generated logical relationships with the logical representation of the data. See *Application*, page 31, line 35 - page 33, line 22; FIG. 9. The program contained in the computer readable medium also includes storing the associations and the generated logical relationships on one or more computer-readable storage media. See *Application*, page 10, lines 13-27; page 31, line 35 - page 33, line 22; FIG. 9.

F. CLAIM 29 - INDEPENDENT

A computer-readable medium containing a program which, when executed by a processor, performs a process of logically representing relationships between data elements defined according to a first physical representation of data. See *Application*, page 5, line 29 – page 6, line 15. As claimed, the program contained in the computer readable medium includes generating a second physical representation of the data from the first physical representation. See *Application*, page 24, lines 18 – page 25, line 12; FIG. 8. The program contained in the computer readable medium also includes generating a logical representation of the data as represented according to the second physical representation, the logical representation abstractly describing the second physical representation of the data. See *Application*, page 24, lines 18 – page 25, line 12; FIG. 8. The program contained in the computer readable medium also includes, on the basis of the relationships between the data elements defined according to the first physical representation of the data, determining corresponding relationships between corresponding data structures defined according to the second physical representation of the data. See *Application*, page 25, line 13 – page 30, line 7; FIG. 8-9. The program contained in the computer readable medium also includes generating logical relationships abstractly describing the determined corresponding relationships. See *Application*, page 25, line 19 – page 31, line 34; FIG. 8-9. The program contained in the computer readable medium also includes including the generated logical relationships with the logical representation, wherein each generated logical relationships describes a path for traversing the second physical representation from a first data structure to a second data structure when processing a query requesting information related to the first and second data structures. See *Application*, page 31, line 35 - page 33, line 22;

FIG. 9. The program contained in the computer readable medium also includes storing the second physical representation, the generated logical relationships, and the logical representation on one or more computer-readable storage media. See *Application*, page 10, lines 13-27; page 31, line 35 - page 33, line 22; FIG. 9.

G. CLAIM 35 - INDEPENDENT

A computer-readable medium containing a program which, when executed by a processor, performs a process of logically representing relationships between data elements described in an eXtended Markup Language (XML) document. See *Application*, page 6, lines 16-31. As claimed, the program contained in the computer readable medium includes retrieving a relational database schema for a plurality of data structures, each data structure corresponding to one of the data elements. See *Application*, page 23, lines 1-12; FIG. 6. The program contained in the computer readable medium also includes retrieving a logical representation abstractly describing the relational database schema. See *Application*, page 24, line 18 – page 25, line 12; FIG. 8. The program contained in the computer readable medium also includes determining the relationships between the data elements from the XML document. See *Application*, page 25, lines 13-26; FIG. 8. The program contained in the computer readable medium also includes, on the basis of the determined relationships, determining corresponding relationships between corresponding data structures defined according to the relational database schema. See *Application*, page 25, line 13 – page 30, line 7; FIG. 8-9. The program contained in the computer readable medium also includes generating logical relationships abstractly describing the determined corresponding relationships. See *Application*, page 25, line 19 – page 31, line 34; FIG. 8-9. The program contained in the computer readable medium also includes including the generated logical relationships with the logical representation, where each of the generated logical relationships describes a path for traversing a relational database constructed according to the relational database schema from a first data structure to a second data structure when processing a query requesting information related to the first and second data structures. See *Application*, page 31, line 35 - page 33, line 22; FIG. 9. The program contained in the computer readable medium also includes storing

the generated logical relationships on one or more computer-readable storage media. See *Application*, page 10, lines 13-27; page 31, line 35 - page 33, line 22; FIG. 9.

H. CLAIM 36 - INDEPENDENT

A computer-readable medium containing a program which, when executed by a processor, performs a process of querying physical data logically represented by a data abstraction model, wherein the physical data being queried is contained in data structures generated from a data source having a different schema from the data structures containing the physical data being queried. See *Application*, page 7, lines 1-13. As claimed, the program contained in the computer readable medium includes receiving an abstract query comprising logical fields and corresponding values, wherein each of the logical fields is defined in the data abstraction model and wherein one or more of the logical fields are result fields to be returned by execution of the abstract query. See *Application*, page 13, lines 15-28; FIG. 2. The program contained in the computer readable medium also includes transforming, by operation of a processor, the abstract query into an executable query capable of being executed against the physical data; wherein the transforming is done using the data abstraction model and wherein the data abstraction model defines a specific path for traversing the data structures containing the physical data to reach the one or more result fields. See *Application*, page 13, line 29 – page 17, line 23; FIG. 2; page 25, line 19 – page 31, line 34; FIG. 8-9.

I. CLAIM 44 - INDEPENDENT

Claim 44 recites an article. See *Application*, page 7, lines 14-23. As claimed, the article includes a data structure comprising: a plurality of logical field specifications, each abstractly describing at least one of a plurality of data structures defined according to a physical representation of data, wherein at least one of the plurality of logical field specifications includes one or more logical relationships algorithmically generated from relationship information describing relationships between the data represented according to another physical representation of the data, each logical relationship describing a path for traversing the physical representation of the data from a first data

structure to a second data structure when processing a query requesting information related to the first and second data structures. See *Application*, page 24, line 18 – page 31, line 34; FIG. 8-9. The article also includes a computer-readable storage medium containing the data structure. See *Application*, page 10, lines 13-27; page 31, line 35 - page 33, line 22; FIG. 9.

Grounds of Rejection to be Reviewed on Appeal

1. Rejection of claims 1-3, 5-7, 9-12, 14-16, 18-21, 23-25, 27-30, 32-34, 36-38, 40-42, and 44 under 35 U.S.C. § 102(b) as being anticipated by *Depledge et al.* (U.S. 5,899,988, hereinafter *Depledge*).

2. Rejection of claims 4, 8, 13, 17, 22, 26, 31, 35, 39, and 43 under 35 U.S.C. § 103(a) as being unpatentable over *Depledge* as applied to claims above, and in view of *Murthy et al* (U.S. Publication 2004/0220927 A1, hereinafter *Murthy*).

ARGUMENTS

1. Claims 1-3, 5-7, 9-12, 14-16, 18-21, 23-25, 27-30, 32-34, 36-38, 40-42, and 44 are not anticipated by *Depledge* under 35 U.S.C. § 102(b).

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

Applicants respectfully submit that *Depledge* does not disclose "each and every element as set forth in the claim". For example, *Depledge* does not disclose a method of logically representing relationships between data elements that includes the act of "*providing a logical representation of the data, the logical representation abstractly describing a second physical representation of the data, wherein the second physical representation of the data is generated from the first physical representation of the data*," as recited in claim 1. Independent claim 11 includes a similar limitation.

Regarding this first element of claim 1, the Examiner states in Final Office Action, p.17:

> With respect to Applicant's argument, the argument is not correct and Examiner is not persuaded because Depledge clearly provides a logical representation of the data (*see Figure 2A; Note that the "Bitmap" column logically represents the data in that each bit logically corresponds to the "customer#" column in Figure 1.*), the logical representation abstractly describes a second physical representation of the data (*see Figure 2A; Note that the "Key" column is a second representation in that it corresponds to all of the possible values for location as*

> *shown in Figure 1 under the "Location" column. The "Bitmap" column logically describes the "Key" column by identifying which customer#s correspond to a given location.*), wherein the second physical representation of the data is generated from the first physical representation of the data (*The "Key" column is generated based on the data portrayed in Figure 1. The "Key" column corresponds to the locations within the data table (100) of Figure 1. By looking at the bitmapped index (200), one can see how each location logically corresponds to each customer#.*).
>
> The second physical representation is distinct from the logical representation in that the "Key" column represents the possible values for a location in data table (100), whereas the "Bitmap" column represents the location of each customer# in data table (100). Therefore, both representations are two totally different/distinct relationships.

In this statement, the Examiner draws an analogy between the present claims and *Depledge.* More specifically, the Examiner argues that the "Bitmap" columns of the bitmapped index 200 (*i.e.*, a "logical representation") abstractly describe the "Key" column of the bitmapped index 200 (*i.e.*, a "second physical representation of the data"). Further, the Examiner asserts that the "Key" column of the bitmapped index 200 (*i.e.*, a "second physical representation of the data") is generated from "data table (100) of Figure 1" (*i.e.*, a "first physical representation of the data"). Thus, the Examiner's analogy to the recited elements can be illustrated as follows:

PRESENT CLAIM

DEPLEDGE

First Physical Representation Of Data

used to generate:

Second Physical Representation Of Data

abstractly described by:

Logical Representation Of Data

Depledge Fig. 1 – Data Table 100

CUSTOMER #	LOCATION	TYPE
101	EAST	BUSINESS
102	NORTH	INDIVIDUAL
103	WEST	INDIVIDUAL
104	EAST	BUSINESS
105	SOUTH	BUSINESS
106	NORTH	INDIVIDUAL

Depledge Fig. 2 – Bitmapped Index 200

KEY	BITMAP
NORTH	0 1 0 0 0 1
SOUTH	0 0 0 0 1 0
EAST	1 0 0 1 0 0
WEST	0 0 1 0 0 0

Applicants respectfully disagree with the Examiner's use of two columns of the bitmapped index 200 as analogous to the "logical representation" and the "second physical representation," which are claimed as distinct elements. Applicants submit that the Bitmap and Key columns are integral parts of the bitmapped index 200 (*Depledge* Fig. 2), and cannot be properly considered to constitute two different representations of the same data. For example, the Bitmap column, if isolated from the Key column, has no discernable meaning whatsoever, and does not "represent" anything. Thus, even

assuming the index 200, as a whole, may be considered a representation of data, the constituent parts of the index cannot be understood as *distinct* representations of data.

In this regard, Applicants respectfully point out that the Examiner's analysis fails to appreciate the distinction between a representation of data, and the data itself, as is required by the claims. For example, claim 1 recites "a first physical representation of data" and "a second physical representation of data". Claim 1 further recites "data elements defined according to [the] first physical representation of data". This language makes clear that the first and second physical representations define the representation of the data, but are not themselves the data. Common examples of physical data representations include document type definitions (DTD) and schemas. The Examiner's analysis focuses on the actual physical tables and their respective data, not on data representations of the data (*i.e.* the schemas of the tables and indexes). Further, there is no suggestion that the data structures relied upon by the Examiner are defined according to different schemas. In fact, quite the opposite, the data structures (*i.e.* the table 100 and the index 200) are both tables that conform to a relational database schema. Therefore, the Examiner's analogy is invalid, and applicants respectfully request that the rejection be withdrawn.

However, even assuming, *arguendo*, that the Bitmap and Key columns are considered as distinct "representations," the Examiner's analogy fails to conform to the remaining limitations of claim 1. For example, the second element of claim 1 recites: "*on the basis of the relationships between the data elements defined according to the first physical representation of the data, determining corresponding relationships between corresponding data structures defined according to the second physical representation of the data*." Applying the Examiner's analogy, this statement would be translated as:

> on the basis of the relationships between the data elements defined according to the Data Table 100 ("first physical representation of data"), determining corresponding relationships between corresponding data structures defined according to the Key Column ("second physical representation of data") of the Bitmapped Index 200.

Applicants submit that this translated statement makes no sense. *Depledge* does not disclose any data structures of any sort that are "defined according to the Key Column." Even if we assume that the values of the Key Column (*i.e.*, "NORTH", "SOUTH", "EAST", "WEST") are data structures, there is no relationship between these values that is determined "on the basis of the relationships between data elements defined according to the Data Table 100." Clearly, the Examiner's analogy does not conform to this limitation, and is thus invalid.

The third element of claim 1 recites: "*generating logical relationships abstractly describing the determined corresponding relationships, each logical relationship defining a path between data structures of the second physical representation*." Independent claims 11, 19, 29, 35 and 44 include similar limitations. Applying the Examiner's analogy, this element of claim 1 would be translated as:

> generating logical relationships abstractly describing the determined corresponding relationships, each logical relationship defining a path between data structures of the Key Column ("second physical representation") of the Bitmapped Index 200.

As described above, the previous step of "determining corresponding relationships" makes no sense as applied to the Examiner's analogy. Thus, there can be no "determined corresponding relationships," as required by this step. However, even ignoring this problem, *arguendo*, the Examiner's analogy also fails to generate a logical relationship defining a path between data structures of the Key Column. Even if we assume that the data structures are the values of the Key Column (*i.e.*, "NORTH", "SOUTH", "EAST", "WEST"), it makes no sense to define a "path" between these values. Again, the Examiner's analogy does not conform to this limitation, and is thus invalid.

Turning now to claim 18: *Depledge* does not teach "receiving an abstract query comprising logical fields and corresponding values, wherein each of the logical fields is defined in the data abstraction model," and "transforming, by operation of a processor,

the abstract query into an executable query capable of being executed against the physical data," as recited in claim 18.

In rejecting claim 18, the Examiner argues in Final Office Action, p. 8:

> Depledge further teaches transforming the abstract query into an executable query capable of being executed against the physical data *(see Fig. 1 and 2A and column 3, lines 22-44; Note that once the location was changed, the values were then changed in the physical data using an executable query which flipped the bits representing the new changed values.);* wherein the transforming is done using the data abstraction model and wherein the data abstraction model defines a specific path for traversing the data structures containing the physical data to reach the one or more result fields *(see Fig. 3 and column 3, lines 22-44; Note that a bitmap index such as the one showed in Fig 3 (302) is used to map each bit for a given location to the physical data (shown in Fig. 1). The result index (320) represents the bitmap of the physical data as a result of the query terms portrayed in Fig. 3 (i.e TYPE = 'BUSINESS' and LOCATION = 'EAST' or LOCATION = 'SOUTH').).*

Independent claim 36 is rejected on a similar basis. Here, the Examiner asserts that the bitmapped indexes of *Depledge* (*e.g.*, items 200 and 300) teach the recited element of a data abstraction model. However, following the Examiner's analogy, this would require the step of "receiving an abstract query comprising logical fields," wherein the logical fields are defined in a bitmapped index (*i.e.*, "data abstraction model"). Applicants submit that the bitmapped indexes of *Depledge* do not define logical fields, or any other type of field.

Further, *Depledge* does not disclose the recited step of "transforming, by operation of a processor, the abstract query into an executable query capable of being executed against the physical data," or any other type of transformation of one query into another. The Examiner states "*Note that once the location was changed, the values were then changed in the physical data using an executable query which flipped the bits.*" Clearly, the "bits" do not constitute an abstract query, a logical query, or any kind of query. Therefore, Applicants submit that "flipping the bits" does not teach transforming an abstract query into an executable query, but rather teaches

transforming bits within the bitmapped index, *i.e.*, the “data abstraction model” according to the Examiner’s analogy.

For the foregoing reasons, applicants respectfully submit that *Depledge* does not teach “each and every element” of the recited claim. Therefore, the claims are believed to be allowable, and allowance of the claims is respectfully requested.

2. Claims 4, 8, 13, 17, 22, 26, 31, 35, 39, and 43 are not unpatentable over *Depledge* as applied to claims above, and in view of *Murthy*, under 35 U.S.C. § 103(a).

The Examiner bears the initial burden of establishing a *prima facie* case of obviousness. *See* MPEP § 2142. To establish a *prima facie* case of obviousness three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one ordinary skill in the art to modify the reference or to combine the reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *See* MPEP § 2143. The present rejection fails to establish at least the third criterion.

Claims 4, 8, 13, 17, 22, 26, 31, 35, 39, and 43 depend, directly or indirectly, on claims that are believed to be allowable, for reasons discussed above. Therefore, the present rejection does not establish a *prima facie* case of obviousness at least because the rejection does not teach or suggest all the claim limitations. Accordingly, Applicants submit these claims are also allowable and respectfully request withdrawal of this rejection.

CONCLUSION

The Examiner errs in finding that:

1. Claims 1-3, 5-7, 9-12, 14-16, 18-21, 23-25, 27-30, 32-34, 36-38, 40-42, and 44 are anticipated by *Depledge*; and
2. Claims 4, 8, 13, 17, 22, 26, 31, 35, 39, and 43 are unpatentable over *Depledge* in view of *Murthy*.

Withdrawal of the rejections and allowance of all claims is respectfully requested.

Respectfully submitted, and
S-signed pursuant to 37 CFR 1.4,

/Gero G. McClellan, Reg. No. 44,227/

Gero G. McClellan
Registration No. 44,227
Patterson & Sheridan, L.L.P.
3040 Post Oak Blvd., Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Appellants

CLAIMS APPENDIX

1. (Previously Presented) A computer-implemented method of logically representing relationships between data elements defined according to a first physical representation of data, comprising:

providing a logical representation of the data, the logical representation abstractly describing a second physical representation of the data, wherein the second physical representation of the data is generated from the first physical representation of the data;

on the basis of the relationships between the data elements defined according to the first physical representation of the data, determining corresponding relationships between corresponding data structures defined according to the second physical representation of the data;

generating logical relationships abstractly describing the determined corresponding relationships, each logical relationship defining a path between data structures of the second physical representation;

associating the generated logical relationships with the logical representation of the data; and

storing the associations and the generated logical relationships on one or more computer-readable storage media.

2. (Original) The method of claim 1, wherein the logical representation comprises a plurality of logical field specifications, and wherein associating comprises including the generated logical relationships with respective logical field specifications.

3. (Original) The method of claim 1, wherein the first physical representation of the data is a document in text-based markup language.

4. (Original) The method of claim 3, wherein the text-based markup language is one of the eXtended Markup Language (XML) and the MicroArray Gene Expression Markup Language (MAGE-ML).

5. (Original) The method of claim 1, wherein the second physical representation is a relational representation.

6. (Original) The method of claim 5, wherein each data structure is a table of the relational representation.

7. (Original) The method of claim 1, wherein the first physical representation is a hierarchical representation and the second physical representation is a relational representation.

8. (Original) The method of claim 7, wherein the hierarchical representation is the eXtended Markup Language (XML).

9. (Original) The method of claim 7, wherein each data structure is a table of the relational representation.

10. (Original) The method of claim 1, further comprising removing any redundant determined corresponding relationships before generating the logical relationships.

11. (Previously Presented) A computer-implemented method of logically representing relationships between data elements defined according to a first physical representation of data, comprising:

generating a second physical representation of the data from the first physical representation;

generating a logical representation of the data as represented according to the second physical representation, the logical representation abstractly describing the second physical representation of the data;

on the basis of the relationships between the data elements defined according to the first physical representation of the data, determining corresponding relationships between corresponding data structures defined according to the second physical representation of the data;

generating logical relationships abstractly describing the determined corresponding relationships;

including the generated logical relationships with the logical representation; wherein each of the generated logical relationships describes a path for traversing the second physical representation from a first data structure to a second data structure when processing a query requesting information related to the first and second data structures; and

storing the second physical representation, the generated logical relationships, and the logical representation on one or more computer-readable storage media.

12. (Original) The method of claim 11, wherein the first physical representation of the data is a document in text-based markup language.

13. (Original) The method of claim 12, wherein the text-based markup language is one of the eXtended Markup Language (XML) and the MicroArray Gene Expression Markup Language (MAGE-ML).

14. (Original) The method of claim 11, wherein the second physical representation is a relational representation.

15. (Original) The method of claim 14, wherein each data structure is a table of the relational representation.

16. (Original) The method of claim 11, further comprising removing any redundant determined corresponding relationships before generating the logical relationships.

17. (Previously Presented) A computer-implemented method of logically representing relationships between data elements described in an eXtended Markup Language (XML) document, comprising:

retrieving a relational database schema for a plurality of data structures, each data structure corresponding to one of the data elements;

retrieving a logical representation abstractly describing the relational database schema;

determining the relationships between the data elements from the XML document;

on the basis of the determined relationships, determining corresponding relationships between corresponding data structures defined according to the relational database schema;

generating logical relationships abstractly describing the determined corresponding relationships;

including the generated logical relationships with the logical representation; wherein each of the generated logical relationships describes a path for traversing a relational database constructed according to the relational database schema from a first data structure to a second data structure when processing a query requesting information related to the first and second data structures; and

storing the generated logical relationships on one or more computer-readable storage media.

18. (Previously Presented) A computer-implemented method of querying physical data logically represented by a data abstraction model, wherein the physical data being queried is contained in data structures generated from a data source having a different schema from the data structures containing the physical data being queried, comprising:

receiving an abstract query comprising logical fields and corresponding values, wherein each of the logical fields is defined in the data abstraction model and wherein one or more of the logical fields are result fields to be returned by execution of the abstract query; and

transforming, by operation of a processor, the abstract query into an executable query capable of being executed against the physical data; wherein the transforming is done using the data abstraction model and wherein the data abstraction model defines

a specific path for traversing the data structures containing the physical data to reach the one or more result fields.

19. (Previously Presented) A computer-readable medium containing a program which, when executed by a processor, performs a process of logically representing relationships between data elements defined according to a first physical representation of data, the process comprising:

retrieving a logical representation of the data, the logical representation abstractly describing a second physical representation of the data, wherein the second physical representation of the data is generated from the first physical representation of the data;

on the basis of the relationships between the data elements defined according to the first physical representation of the data, determining corresponding relationships between corresponding data structures defined according to the second physical representation of the data;

generating logical relationships abstractly describing the determined corresponding relationships, each logical relationship defining a path between data structures of the second physical representation;

associating the generated logical relationships with the logical representation of the data; and

storing the associations and the generated logical relationships on one or more computer-readable storage media.

20. (Original) The computer-readable medium of claim 19, wherein the logical representation comprises a plurality of logical field specifications, and wherein associating comprises including the generated logical relationships with respective logical field specifications.

21. (Original) The computer-readable medium of claim 19, wherein the first physical representation of the data is a document in text-based markup language.

22. (Original) The computer-readable medium of claim 21, wherein the text-based markup language is one of the eXtended Markup Language (XML) and the MicroArray Gene Expression Markup Language (MAGE-ML).

23. (Original) The computer-readable medium of claim 19, wherein the second physical representation is a relational representation.

24. (Original) The computer-readable medium of claim 23, wherein each data structure is a table of the relational representation.

25. (Original) The computer-readable medium of claim 19, wherein the first physical representation is a hierarchical representation and the second physical representation is a relational representation.

26. (Original) The computer-readable medium of claim 25, wherein the hierarchical representation is the eXtended Markup Language (XML).

27. (Original) The computer-readable medium of claim 25, wherein each data structure is a table of the relational representation.

28. (Original) The computer-readable medium of claim 19, wherein the process further comprises:

removing any redundant determined corresponding relationships before generating the logical relationships.

29. (Previously Presented) A computer-readable medium containing a program which, when executed by a processor, performs a process of logically representing relationships between data elements defined according to a first physical representation of data, the process comprising:

generating a second physical representation of the data from the first physical representation;

generating a logical representation of the data as represented according to the second physical representation, the logical representation abstractly describing the second physical representation of the data;

on the basis of the relationships between the data elements defined according to the first physical representation of the data, determining corresponding relationships between corresponding data structures defined according to the second physical representation of the data;

generating logical relationships abstractly describing the determined corresponding relationships; and

including the generated logical relationships with the logical representation; wherein each of the generated logical relationships describes a path for traversing the second physical representation from a first data structure to a second data structure when processing a query requesting information related to the first and second data structures; and

storing the second physical representation, the generated logical relationships, and the logical representation on one or more computer-readable storage media.

30. (Original) The computer-readable medium of claim 29, wherein the first physical representation of the data is a document in text-based markup language.

31. (Original) The computer-readable medium of claim 30, wherein the text-based markup language is one of the eXtended Markup Language (XML) and the MicroArray Gene Expression Markup Language (MAGE-ML).

32. (Original) The computer-readable medium of claim 29, wherein the second physical representation is a relational representation.

33. (Original) The computer-readable medium of claim 32, wherein each data structure is a table of the relational representation.

34. (Original) The computer-readable medium of claim 29, wherein the process further comprises:

removing any redundant determined corresponding relationships before generating the logical relationships.

35. (Previously Presented) A computer-readable medium containing a program which, when executed by a processor, performs a process of logically representing relationships between data elements described in an eXtended Markup Language (XML) document, the process comprising:

retrieving a relational database schema for a plurality of data structures, each data structure corresponding to one of the data elements;

retrieving a logical representation abstractly describing the relational database schema;

determining the relationships between the data elements from the XML document;

on the basis of the determined relationships, determining corresponding relationships between corresponding data structures defined according to the relational database schema;

generating logical relationships abstractly describing the determined corresponding relationships;

including the generated logical relationships with the logical representation; wherein each of the generated logical relationships describes a path for traversing a relational database constructed according to the relational database schema from a first data structure to a second data structure when processing a query requesting information related to the first and second data structures; and

storing the generated logical relationships on one or more computer-readable storage media.

36. (Previously Presented) A computer-readable medium containing a program which, when executed by a processor, performs a process of querying physical data logically represented by a data abstraction model, wherein the physical data being

queried is contained in data structures generated from a data source having a different schema from the data structures containing the physical data being queried, the process comprising:

receiving an abstract query comprising logical fields and corresponding values, wherein each of the logical fields is defined in the data abstraction model and wherein one or more of the logical fields are result fields to be returned by execution of the abstract query; and

transforming, by operation of a processor, the abstract query into an executable query capable of being executed against the physical data; wherein the transforming is done using the data abstraction model and wherein the data abstraction model defines a specific path for traversing the data structures containing the physical data to reach the one or more result fields.

37. (Original) The computer-readable medium of claim 36, wherein the specific path is derived from relationships in the data source.

38. (Original) The computer-readable medium of claim 36, wherein the data source is a document in text-based markup language.

39. (Original) The computer-readable medium of claim 38, wherein the text-based markup language is one of the eXtended Markup Language (XML) and the MicroArray Gene Expression Markup Language (MAGE-ML).

40. (Original) The computer-readable medium of claim 36, wherein the data structures containing the physical data being queried are arranged according to a relational schema.

41. (Original) The computer-readable medium of claim 40, wherein each data structure containing physical data being queried is a database table according to the relational schema.

42. (Original) The computer-readable medium of claim 36, wherein the data source is arranged according to a hierarchical representation and the data structures containing the physical data being queried define a relational representation.

43. (Original) The computer-readable medium of claim 42, wherein the hierarchical representation is the eXtended Markup Language (XML).

44. (Previously Presented) An article, comprising:

a data structure comprising: a plurality of logical field specifications, each abstractly describing at least one of a plurality of data structures defined according to a physical representation of data, wherein at least one of the plurality of logical field specifications includes one or more logical relationships algorithmically generated from relationship information describing relationships between the data represented according to another physical representation of the data, each logical relationship describing a path for traversing the physical representation of the data from a first data structure to a second data structure when processing a query requesting information related to the first and second data structures; and

a computer-readable storage medium containing the data structure.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.